Efficient Lightweight Detection and Classification Method for Field-Grown Horticultural Crops

Yaru Huang[1], Hua Zhou[2], Zhongyi Shu[3]*
School of Cyberspace Security, Software Engineering Institute of Guangzhou, Guangzhou, China [1, 2]
Zhongnan University of Economics and Law, Wuhan, Hubei, China[3]

***Abstract*—As the core carrier of human food supply and agricultural economy, manual management in large-scale crop cultivation faces bottlenecks such as low efficiency, high cost, and difficulty in standardization. There is an urgent need for computer vision technology to realize automated detection and growth stage classification. However, most existing algorithms rely on high-performance GPUs for operation, resulting in high hardware costs, which makes it difficult to popularize them in low-end agricultural edge devices (e.g., embedded controllers, low-cost industrial computers). This study proposes a lightweight crop detection and classification model, Lite-CropNet. It builds a neural network architecture based on the CSPDarknet backbone network, designs a concise decoder, and adopts four-scale detection heads to adapt to crop targets of different sizes, balancing high accuracy and lightweight characteristics. Using tomatoes as the experimental object, tests on the TomatOD dataset (simulating real greenhouse environments) show that Lite-CropNet outperforms advanced methods, with a mean Average Precision (mAP)@0.5 of 85.7%. Under the conditions of the GTX 1650 GPU and 640×640 resolution, the Frame Per Second (FPS) reaches 76.9, and the model size is only 4.4M. This neural network model can efficiently complete tomato detection and maturity classification, and its architecture and design can also be transferred to crops such as potatoes and strawberries, providing a cost-effective and highly universal automated solution for agricultural production.**

Keywords*—Computer vision; neural network; object detection and classification; lightweight; horticultural crops

I. Introduction

In large-scale agricultural production, accurate detection and growth stage classification of crops are core links for yield assessment, quality control, and automated management. Whether for vegetable and fruit crops such as tomatoes and potatoes, or food crops such as wheat and corn, real-time monitoring of their growth status directly affects production efficiency and economic benefits. With population growth and the development of agricultural intensification, models such as greenhouses and large-scale field cultivation have been widely applied [1]. However, traditional manual detection methods, limited by low efficiency, long time consumption, and high cost, can hardly meet the management needs of large-scale and multi-variety crops [2], [3].

With the development of computer vision and deep learning technologies, the advancement of image recognition technology in recent years has enabled Convolutional Neural Networks (CNNs) to be widely used in deep learning [4], becoming an important method for image classification, object detection, and recognition. Especially in the agricultural field, deep learning models based on the CNN architecture have been gradually applied [5], [6], [7] due to their high accuracy in image classification and object detection tasks. At this point, how to collect these images for deep learning training is a question that researchers should consider. Currently, commonly used methods include satellite remote sensing, unmanned aerial vehicles (UAVs), fixed platforms, and unmanned vehicles. These devices have their own advantages and disadvantages: satellite remote sensing systems are suitable for long-distance and large-scale monitoring but have relatively low image resolution, making it difficult to capture objects at the small organ level [8]; UAVs are usually used for large-scale field monitoring [9]; these two technologies are suitable for long-distance shooting and have high costs. Fixed platforms are installed at fixed positions and are more suitable for monitoring small-scale areas. In contrast, unmanned vehicle technology has shown excellent performance in large-scale farmland or greenhouses—it can not only adapt to the detection of underlying fruits of hanging crops such as tomatoes but also meet the close-range image collection of low-growing crops such as potatoes, providing data support for high-precision detection of multi-variety crops.

However, existing crop detection and classification technologies still face two core challenges: First, insufficient scene adaptability—different crops have significant differences in growth forms (e.g., hanging tomatoes, creeping potatoes, clustered strawberries) and appearance features (color, shape, size). In addition, the field environment has interferences such as light changes and occlusions (overlapping of leaves, branches, and fruits), making it difficult for a single model to adapt to multi-variety crops. Second, high hardware dependence—most existing models require high-performance GPUs for support, with large parameter scales and high computational costs, which cannot be deployed in low-end agricultural edge devices (e.g., embedded controllers, low-cost industrial computers), limiting the large-scale application of multi-variety crop detection technologies. Bathini and Usha Rani [10] also pointed out in their review on the application of artificial intelligence in agricultural crop yield analysis that although current deep learning models perform well in crop management, yield prediction, and other fields, they generally have problems of high dependence on data quality and insufficient adaptability of lightweight models.

To address this challenge, this study adopts the TomatOD dataset [11], a highly specialized and innovative dataset designed specifically for tomato fruit object detection and

*Corresponding author.

classification. Collected in a real greenhouse environment, it includes low-light conditions and long-distance shooting scenarios, providing challenging scenes for the model and helping it better adapt to real-world applications. Against this background, the study proposes the Lite-CropNet model, whose core differences in mathematical architecture from existing methods are as follows: 1) Feature extraction: A multi-stage structure of $Ck(m) - Pk(m)$ is adopted to output multi-scale feature maps with 32/64/128/256 channels; 2) Loss function: Smoothed Intersection over Union (SIoU) loss (including angle cost $\Delta = \sum_{t=x,\gamma} (1 - e^{-}{}_{\gamma\varrho t})$ and shape cost $\Omega = \sum_{t=} w,h\ (1 - e^{-w}{}_{t})^{th}$) is introduced; 3) Lightweight: The parameter computation amount is controlled to 4.4M according to the formula $Params = i \cdot (k \cdot k) \cdot o + o$ (where i is input size, k is convolution kernel, o is output size), which is only 2.8% of the DETR model (158M), solving the problem of high hardware dependence. While being lightweight, Lite-CropNet maintains high accuracy, with a mean Average Precision (mAP) of 85.7%. Notably, Lite-CropNet has only 4.4M parameters, and under the image resolution of 640×640, it achieves an excellent frame rate of 76.9 Frames Per Second (FPS) on the low-cost GTX 1650 GPU. This enables Lite-CropNet to have wide applicability in real-time applications and provides a cost-effective and efficient automated solution for agricultural production.

In summary, the main contributions of this study are threefold: 1) Lite-CropNet: a novel deep convolutional network with a lightweight encoder and a decoder that effectively processes encoded features, alleviating the problems of information loss and degradation. 2) Innovative use of four detection heads of different scales, which not only has excellent detection performance for medium and large target crops such as tomatoes but also can accurately identify other small clustered crops, improving the model's crop variety adaptability. 3) High real-time efficiency is reported on low-cost devices, providing an effective and economical solution for the application of modern high-throughput plant phenotyping platforms.

The structure of this study is arranged as follows: Section I (this section) introduces the research background. Section II presents related work by researchers and highlights the problem statement. Section III provides a detailed description of the Lite-CropNet model. Section IV introduces dataset details, challenges, and experimental design. Section V conducts experiments and performs comparative analysis with other models from multiple dimensions. Section VI discusses the research. Section VII draws the conclusions of this research and proposes future work.

II. Related Work

In recent years, deep learning models based on Convolutional Neural Networks (CNNs) have been widely used in agricultural crop detection and classification tasks. Researchers have carried out explorations around "improving accuracy", "adapting to scenes", and "optimizing efficiency", forming research results in multiple directions. In terms of crop and weed detection and classification, Ahmad et al. [12] proposed a detection model based on Vision Transformer (ViT). Through pixel-level annotation training of high-resolution UAV images, it overcomes the problem of similar appearance between crops and weeds, achieving a classification accuracy of 89.4%, which is significantly better than traditional models such as UNet and Fully Convolutional Network (FCN). Zhang et al. [13] further proposed the LMS-YOLO11n lightweight multi-scale weed detection model. Through the design of FastGLU feature extraction module and Attention Hierarchical Feature Pyramid Network (AHFPN) feature fusion network, the mAP is increased by 2.5% on the CottonWeedDet3 dataset, while the model parameters and computational complexity are reduced by 37% and 26% respectively. Reedha et al. [14] used Transformer neural networks to process high-resolution UAV images, further optimizing the classification accuracy of weeds and crops. López-Correa et al. [15] designed an intelligent weed management system in tomato fields, providing technical support for precise weeding in the field.

In terms of crop organ and maturity detection, Mu et al. [16] proposed a tomato detection model based on Faster R-CNN and ResNet-101. By transferring learning from the COCO dataset, it can automatically detect intact green tomatoes (ignoring occlusion and growth stages), with an average precision of 87.83% on the test set. Su et al. [17] improved the SE-YOLOv3-MobileNetV1 network, introduced a channel attention mechanism, and could distinguish four tomato maturity levels, with an average precision of 87.7%. Rahim et al. [18] used Faster R-CNN to realize tomato flower detection and counting in greenhouses. Rahim et al. [19] further quantified the number of grapevine inflorescences and flowers, providing data for yield prediction. Suh et al. [20] realized the classification of sugar beets and potatoes through transfer learning, adapting to field scenarios. Nkemelu et al. [21] used a deep CNN to complete crop seedling classification, solving the problem of seedling stage recognition.

In terms of data collection, commonly used devices include satellite remote sensing, UAVs, fixed platforms, and unmanned vehicles [8], [9]. Remote sensing is suitable for large-scale areas but has low resolution; UAVs have high costs; fixed platforms are suitable for small-scale areas; unmanned vehicles are suitable for image collection of various crops such as tomatoes and potatoes. Tsironis et al. [11], [22] constructed the TomatOD dataset, which contains 277 images and 2,413 tomato samples, annotates three maturity stages, and covers complex scenarios such as light changes and occlusions, with good challenging properties.

Although existing studies have made progress, there are still three major bottlenecks: First, poor scene adaptability—most models rely on ideal datasets and are difficult to cope with complex field environments such as low light and high occlusion. Second, high hardware dependence—models such as DETR [23] and Faster R-CNN [24] have large parameter sizes (DETR reaches 158M), making it difficult to deploy them on agricultural edge devices. Third, weak multi-variety adaptability—existing studies mostly focus on a single crop or task, and do not fully consider the morphological differences between crops (e.g., hanging tomatoes, clustered strawberries), making it difficult to achieve universal detection.

To address the above problems, this study proposes Lite-CropNet, achieving the following breakthroughs: In terms of lightweight design, through multi-stage feature extraction and a concise decoder, the number of parameters is controlled at 4.4M (only 2.8% of DETR), adapting to low-end devices such as GTX 1650 and J4125. In terms of scene robustness, training is conducted based on the TomatOD dataset, and the SIoU loss function (including angle and shape costs) is introduced to improve adaptability to complex field environments. In terms of multi-variety detection capability, an innovative four-scale detection head is designed to support various crops such as tomatoes (medium and large-sized) and strawberries (small clustered), reducing the cost and complexity of multi-variety deployment.

III. Materials and Methods

This section provides a detailed introduction to the design of the Lite-CropNet deep learning framework for crop detection and classification.

A. Lite-CropNet Model Design

In crop detection and classification, the model is required to have higher robustness due to the influence of various natural factors and imaging from different shooting angles. Moreover, considering the deployment needs of edge devices in the context of plant science, the model is also required to be lightweight. Therefore, in the model architecture design, both aspects are taken into account: while reducing parameters, the performance of the model is maintained, and the Lite-CropNet model is designed, which consists of three parts. The Encoder (feature extraction network) is responsible for extracting image features; after feature enhancement of the feature maps, the Decoder combines, utilizes, and decodes the features from the Encoder, aggregates low-level spatial features and high-level semantic features, and improves the recognition accuracy of objects of different scales. Finally, the features are transmitted to the Detector for detection, generating object bounding boxes with coordinates, categories, and confidence levels. The overall architecture of the Lite-CropNet model is shown in Fig. 1, and the global architecture of Lite-CropNet and its optimization are introduced below.

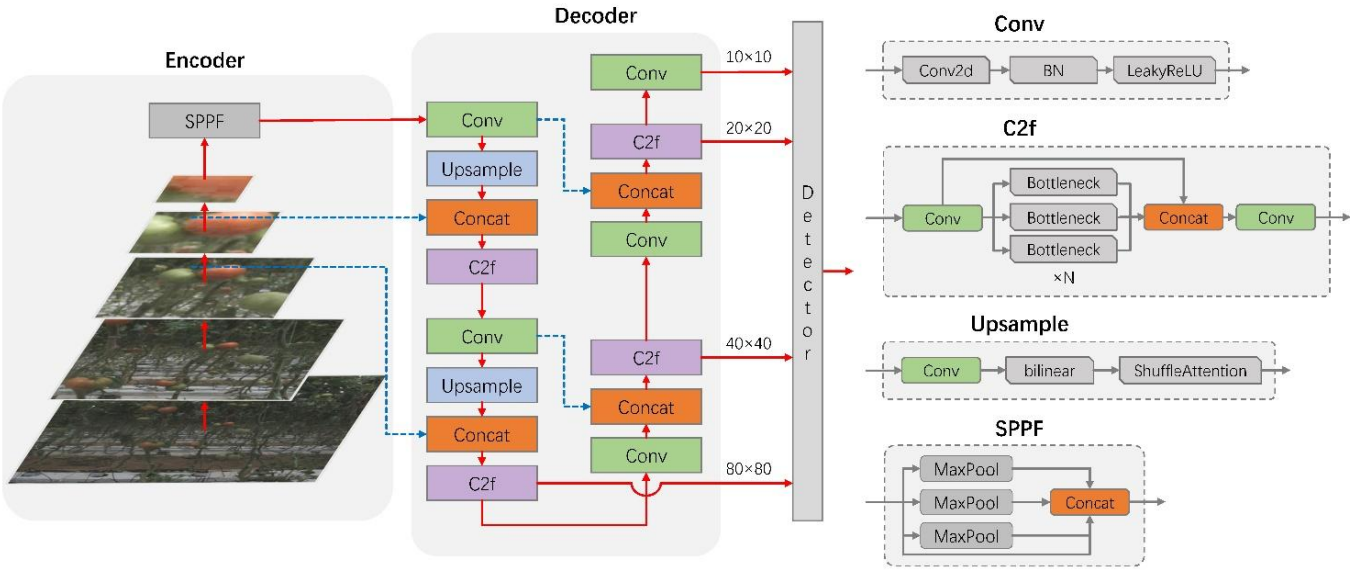

Fig. 1. Lite-CropNet model structure and module details.

1) Encoder: For the encoder part, the general structure of Cross Stage Partial Darknet (CSPDarknet) [25] is adopted and its design is retained. First, an RGB image $I \in R^{H\times W\times 3}$ is given as input, where H and W represent the height and width of the output feature map, respectively. The entire Encoder is composed of 5 convolutional layers and 4 feature extraction layers, specifically defined as Eq. (1):

$$C3(16) - C3(32) - P1(32) - C3(64) - P2(64) - C3(128) - P3(128) - C3(256) - P4(256) \quad (1)$$

Here, $Ck(m)$ denotes a 2D convolutional layer with m channels and $k \times k$ filters, all with a stride of 2; $Pk(m)$ denotes the k^{th} feature extraction layer that outputs m channels. By inserting CSPLayer [26] at different stages to extract features, 32, 64, 128 and 256 channels are output, generating feature maps at different stages. After the last feature extraction layer (P4), a Spatial Pyramid Pooling - Fast (SPPF) [27] module is introduced. The number of input channels of this module is equal to the number of output channels of the last feature layer ($C_{input} = 256$), and the kernel size is $K_{SPPF} = 5 \times 5$. Without changing the size of the feature map, pooling is performed on the feature map at different scales, which helps the network better capture the semantic information of various targets.

2) Decoder: In the design of the Decoder, first, a convolutional layer with a kernel size of 1×1 is used to compress the 256-channel feature map extracted by the SPPF in the Encoder to 128 channels, reducing computational complexity while retaining key features. Subsequently, upsampling is performed to increase the spatial resolution of

the feature map, which is then concatenated with the feature map of P4. A Cross Stage Partial Fusion (C2f) module [28] is introduced to optimize the feature representation. Next, the same operation is carried out again, and the result is concatenated with the feature map of P3. Finally, a convolution with a stride of 2 and a kernel size of 3×3 is further connected, and the result is concatenated with the previous feature maps to form a more optimized feature representation. In this process, emphasis is placed on the utilization of P2 and P3 feature maps because they have a suitable scale for object detection.

Another important design of the Lite-CropNet network is to add a prediction head to detect ultra-large targets while simplifying the process of adding detection layers. During the inspection of the dataset, it was found that some images contain ultra-large crop targets, which also exist in practical applications. Therefore, a convolution module with a stride of 2 and a kernel size of 3×3 is directly added after the third feature map passed to the head. This innovation effectively improves the detection ability of ultra-large targets without significantly affecting the overall computational complexity of the model. Through gradual feature fusion and representation optimization (without significantly affecting the overall computational complexity of the model), the decoder finally transmits four feature maps of different sizes to the detection module (Detector) to output the final object detection results. The design of the Decoder has clear objectives in each step of the operation. From channel compression, upsampling to feature concatenation, all are aimed at comprehensively utilizing the sufficiently deep encoded feature layers in the encoder to respond to more abstract information, and combining an adaptive strategy to restore spatial resolution and improve the detection accuracy of targets of different scales.

B. Activation Function

The selection of the activation function is of great significance in neural networks. It can introduce non-linear factors into the neural network and improve the model's expressive ability. In the Lite-CropNet model, attempts were made to use SiLU [29], ReLU [30], and LeakyReLU [31] activation functions, and their performances were compared. Finally, the more optimal LeakyReLU activation function was selected.

SiLU is a smooth and differentiable activation function with non-monotonicity, which can more easily capture complex patterns. Its non-linear definition is Eq. (2):

$$SiLU(x) = x \cdot \sigma(x) \tag{2}$$

In contrast, ReLU is widely used in neural networks and has simple computation, which helps to improve the computation speed of the network. It can effectively alleviate the problem of gradient disappearance and introduce sparse activation during the training process, making the network easier to optimize. Its non-linear definition is Eq. (3):

$$ReLU(x) = max(0, x) \tag{3}$$

where, x is the input. ReLU can effectively alleviate the problem of gradient disappearance and introduce sparse activation during the training process, making the network easier to optimize. However, it should be noted that ReLU has the problem of neuron death, that is, the output is zero when the input is negative, resulting in some neurons not being activated. To solve the dead neuron problem of ReLU, LeakyReLU introduces a small negative slope for negative inputs while maintaining simplicity. LeakyReLU retains the advantages of ReLU and provides a stronger gradient signal by allowing small negative gradients. This helps to avoid the problem of gradient disappearance during training while maintaining the computational efficiency of the model. The equation of LeakyReLU is Eq. (4):

$$LeakyReLU(x) = \begin{cases} x\ (x > 0) \\ \alpha x\ (x \leq 0) \end{cases} \tag{4}$$

where, α is a small positive slope, such as $0.1, 0.01$, or even smaller.

In practical tasks, the selection of the activation function usually depends on specific needs and experimental results. In the Lite-CropNet model, considering the need to enhance the learning ability, LeakyReLU was finally selected as the activation function because it solves the dead neuron problem, has a lower computational cost compared to SiLU, and has a more stable training process compared to ReLU.

C. Attention Mechanism

The attention mechanism is widely used in computer vision technology. It is a cognitive process that selectively acts on relevant information, enabling the model to ignore redundant information and focus more on useful information. Hu et al. [32] proposed the Squeeze-and-Excitation (SE) attention mechanism, which focuses on channel information and can enhance the channel features of the input feature map. However, due to the introduction of the FC fully connected layer, its computational complexity is relatively high. Woo et al. [33] proposed the Convolutional Block Attention Module (CBAM) attention module. Compared with the SE module, CBAM adds a spatial attention mechanism and focuses on regions of interest. The Efficient Channel Attention (ECA) attention mechanism [34] with global context self-attention generates weights by using global context information, allowing the model to better capture context information, but its computational complexity is relatively high. Shuffle Attention (SA) [35] generates context information by rearranging input features, enhances the robustness of the model to spatial changes, and optimizes the allocation of attention weights through feature grouping and rearrangement. Its core equation is Eq. (5):

$$F_{SA} = Shuffle(Conv2(Conv1(F_{in}, \frac{C}{g}))) \tag{5}$$

where, F_{in} is the input feature map, C is the number of input channels (such as 128, 64), g is the number of groups, $Conv1$ is a 1×1 convolution (compressing the channels to $\frac{C}{g}$), $Conv2$ is a 3×3 convolution (restoring the channels to C), and $Shuffle(\cdot)$ is the feature rearrangement operation. Yu et al. [36] proposed the Mlt_ECA attention module, which is a dimension-free local cross-channel interaction strategy. It generates feature weights by performing 1D convolution

operations, and the convolution kernel size K is adaptively determined by the mapping of the channel dimension C, which is defined as Eq. (6):

$$K = \left|\frac{\log_2(c)+\beta}{\alpha}\right|_{odd} \quad (6)$$

α and β are adjustable hyperparameters, and "odd" indicates that K takes an odd number.

Considering that most of the fruit target areas in the dataset are small and there is a lot of background, which is not helpful for the detection and classification of crop fruits. Therefore, to reduce the impact of the background and pay more attention to the required feature information, during the design process of the Lite-CropNet model, attempts were made to compare the effects of different attention mechanisms and apply them to different stages of the model. A large number of experiments were conducted, and it was found that the experimental results were better when the Shuffle Attention mechanism was applied to the upsampling and downsampling modules. The performance differences of different attention mechanisms will be verified in the ablation experiments of the article.

D. Loss Function

As a metric to calculate the error between the forward propagation result of a neural network in each iteration and the ground truth, the loss function guides the weight adjustment in backpropagation. In the implementation of Lite-CropNet, various commonly used loss functions were tested. For the bounding box loss function, the Complete Intersection over Union (CIoU) loss function was initially considered, with its calculation methods shown in Eq. (7) and Eq. (8):

$$CIoU = IoU - \frac{\rho^2\left(b,b^{g^t}\right)}{c^2} - \alpha v \quad (7)$$

$$L_{CIoU} = 1 - CIoU \quad (8)$$

Intersection over Union (IoU) represents the intersection ratio between the predicted bounding box and the ground truth box; denotes the shortest diagonal length of the minimum enclosing box that contains both the predicted box and the ground truth box; $\rho^2\left(b,b^{g^t}\right)$ is the Euclidean distance between the center of the ground truth box and the center of the predicted box; α is a positive balance parameter; and v represents the consistency of the aspect ratio between the predicted box and the ground truth box. The calculation methods of α and v are shown in Eq. (9) and Eq. (10):

$$v = \frac{4}{\pi^2}\left(arctan\frac{w^{g^t}}{h^{g^t}} - arctan\frac{w}{h}\right)^2 \quad (9)$$

$$\alpha = \frac{v}{(1-IoU)+v} \quad (10)$$

In Eq. (9), h^{g^t} and w^{g^t} represent the height and width of the ground truth box, respectively; h and w represent the height and width of the predicted box, respectively. The CIoU loss function comprehensively considers the overlapping area, aspect ratio, and center distance, and can well measure the relative position between boxes. However, CIoU does not consider the direction matching between the target box and the predicted box, leading to a slow convergence rate. Therefore, this study considered an alternative approach using the Smoothed Intersection over Union (SIoU) loss function [37]. SIoU optimizes the loss calculation by introducing the vector angle between the target box and the predicted box, and it plays an important role in strawberry detection networks through the linear combination of four components: angle cost, distance cost, shape cost, and IoU cost. Its calculation methods are shown in Eq. (11) and Eq. (12):

$$L_{SIoU} = 1 - IoU + \frac{\Delta+\Omega}{2} \quad (11)$$

$$IoU = \frac{|B \cap B^{GT}|}{|B \cup B^{GT}|} \quad (12)$$

where, B and B^{GT} represent the predicted box and the ground truth box, respectively; Ω denotes the shape cost; and Δ represents the distance cost (which incorporates the angle cost and redefines the distance metric). The equations for Ω and Δ are defined as Eq. (13) and Eq. (14):

$$\Omega = \sum_{t=w,h}\left(1 - e^{-w_t}\right)^{\theta} \quad (13)$$

$$\Delta = \sum_{t=x,y}\left(1 - e^{-\gamma\rho_t}\right) \quad (14)$$

In Eq. (13), $w_w = \frac{|w-w^{g^t}|}{max\left(w,w^{g^t}\right)}$, $w_h = \frac{|h-h^{g^t}|}{max\left(h,h^{g^t}\right)}$ and θ represents the degree of attention paid to Ω.

In Eq. (14), $\rho_x = \left(\frac{\left|b_{cx}^{g^t}-b_{c_x}\right|}{c_w}\right)^2$, $\rho_y = \left(\frac{\left|b_{cy}^{g^t}-b_{c_y}\right|}{c_h}\right)^2$, and γ is defined as:

$$p = arcsin\frac{\max\left(b_{cy}^{g^t},b_{c_y}\right)-\min\left(b_{cy}^{g^t},b_{c_y}\right)}{\sqrt{\left(b_{cx}^{g^t}-b_{c_x}\right)^2+\left(b_{cy}^{g^t}-b_{c_y}\right)^2}} - \frac{\pi}{4} \quad (15)$$

$$\gamma = 1 + 2\,sin^2(p) \quad (16)$$

In Eq. (15) and Eq. (16), $b_{c_x}^{g^t}$ and $b_{c_y}^{g^t}$ are the coordinates of the center of the ground truth box; b_{c_x} and b_{c_y} are the coordinates of the center of the predicted box.

The loss function plays a key role in evaluating the performance of a detection model. Typically, at least three loss functions need to be defined: object loss, classification loss, and bounding box loss. The bounding box loss has a significant impact on the detection accuracy and convergence speed of the network model.

To improve the stability of the model in predicting the size and position of target boxes, the SIoU loss function was introduced, which incorporates the vector angle of the expected regression into the calculation of the bounding box regression loss function. The mathematical expression of the SIoU loss function is as follows:

$$L_{SIoU} = \alpha \cdot AngleCost + \beta \cdot DistanceCost + \gamma \cdot ShapeCost + \delta \cdot IoUCost \quad (17)$$

In Eq. (17), α, β, γ and δ are the weight coefficients of the loss terms; $AngleCost$ represents the angle cost; $DistanceCost$ represents the distance

cost; $ShapeCost$ represents the shape cost; and $IoUCost$ represents the IoU cost. Through the linear combination of these four components (angle cost, distance cost, shape cost, and IoU cost), the difference between the predicted box and the ground truth box is measured more comprehensively.

In summary, by introducing the vector angle between the required regression boxes, the SIoU loss function redefines the distance loss, effectively reducing the degree of freedom of regression, accelerating the convergence speed of the network, and further improving the regression accuracy. Therefore, this study adopts the SIoU loss function as the loss function for bounding box regression, which improves the detection accuracy and stability, enabling Lite-CropNet to better complete the classification task of crop fruits of different sizes. This will be further verified in the experiments below.

IV. Experimental Design

In this section, the information of the adopted dataset is first introduced, followed by the presentation of evaluation metrics and experimental details. Additionally, the proposed Lite-CropNet model is compared with advanced detectors on the TomatOD dataset, and their performance is reported. Through the visualization of inference results and the analysis of the model's defects, the characteristics of the model are discussed in depth. Finally, ablation experiments are conducted to verify the selection of the key attention mechanism in Lite-CropNet.

A. Dataset Materials

In this experiment, the TomatOD dataset [11], publicly released by Tsironis et al. (2020) [22], was adopted. The selection basis and dataset characteristics are as follows: This dataset was collected from a real soilless tomato greenhouse on Crete, Greece, aiming to simulate the actual agricultural scenario of robotic arm navigation. Its scene authenticity, task adaptability, and comprehensive challenges are highly consistent with the core goal of this study—"developing a practical lightweight crop detection model". On the one hand, the dataset covers field interferences such as light fluctuations and occlusions by branches, leaves, and fruits, which can verify the model's robustness in complex environments. On the other hand, it contains 277 images and 2,413 tomato samples, which not only annotate accurate bounding boxes but also classify three maturity stages (unripe (green) - semi-ripe (orange-red) - fully ripe (red)) according to color phenotypes.

Since this dataset already provides bounding box annotations—each tomato is labeled with a bounding rectangle and marked with corresponding labels for the three growth stages—these annotated data form the basis for model training. To ensure the accuracy of the annotations, LabelImg [38] was used to check the tomato annotations in the dataset, and at the same time, the characteristics of the dataset were analyzed in depth to help design a better model. The number of tomatoes in each image ranges from 1 to 21, and the size of the annotation box accounts for 3% to 15% of the image size. Tomatoes with a size smaller than a certain range are considered out of scope and not annotated or detected. In terms of the number of category instances, the proportion of labels of different categories in all labels shows that the number of unripe instances is the largest, while the number of semi-ripe instances is the smallest. The TomatOD dataset uses an 8:2 ratio to divide the images into a training set and a test set. The details of the number of images and classification of the dataset are shown in Table I.

TABLE I. Dataset Details

Growth Stage	Training Set Instances	Testing Set Instances	Total	Percentage(Total)
Unripe	1,295	292	1,587	65.8%
Semi-ripe	318	77	395	16.4%
Fully-ripe	332	99	431	17.8%
Total	1,945	468	2,413	100.0%

In addition, this dataset has a unique feature: it is collected in a real greenhouse tomato field. In actual tomato planting scenarios, conditions such as light may change at any time, and the dataset takes this into account, meeting the experimental needs. It includes scenarios such as occlusion by branches, occlusion by tomatoes, occlusion by leaves, and unsuitable dim light, even the superposition of multiple situations. The dataset contains multiple types of occlusion scenarios, and the occlusion degree is quantified using the mathematical equation: $= 1 - \frac{|B_{vis}|}{|B_{gt}|}$. Among them, B_{gt} is the area of the real bounding box of the fruit, B_{vis} is the area of the visible region of the fruit, and O is the occlusion degree ($O \in [0,1]$, where $O = 0$ means no occlusion and $O = 1$ means full occlusion). As shown in Fig. 2, representative images are carefully selected to show the four main challenges in the tomato detection and classification task. For clear display, the brightness of the images has been enhanced, while the actual images have weak light. It is worth noting that to more intuitively display the key parts, these images are shown after magnification. In the actual dataset, the imaging distance of the images is longer, and the tomato targets are smaller.

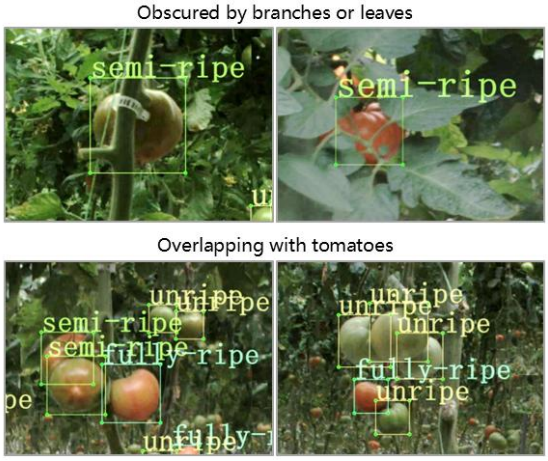

Fig. 2. Main challenges in the TomatOD dataset.

In summary, it can be seen that the many characteristics of the TomatOD dataset increase the difficulty of object detection. Through more realistic scene challenges, it helps the model

better adapt to real-world applications and achieve better robustness and generalization.

B. Experimental Configuration

This experiment was implemented using the PyTorch deep learning framework [39] and accelerated using CUDA. The TomatOD dataset was used for training and evaluation, and all 222 images in the training set were used for training and validation. The image size was adjusted to 640×640 pixels for training, which also meets the resolution requirements when deploying to low-end edge devices. During the training process, Adaptive Moment Estimation (Adam) [40] was used as the optimizer, with the initial learning rate (lr0) and cyclic learning rate (lrf) both set to 0.01, the momentum set to 0.8, and the batch size set to 8 images per batch. The hardware parameters of the equipment used in the experiment were Intel(R) Core(TM) i5-13400F and NVIDIA GeForce GTX 3090 GPU, equipped with a deep neural network acceleration library of CUDA version 11.8, parallel computing framework, and CUDNN version 8.9.5.

After the configuration of relevant parameters was completed, according to the actual convergence of the model, the training process was configured to 280 epochs for optimization. Data augmentation techniques were adopted, including random horizontal flipping, random adjustment of brightness or contrast, and random cropping. The trained model was evaluated using the test set, and each image in the test set was different from those used in the training set. Fig. 3 shows the changes in various losses and mAP values with epochs during the entire training process. It can be observed that the model was properly trained, showing convergence without overfitting.

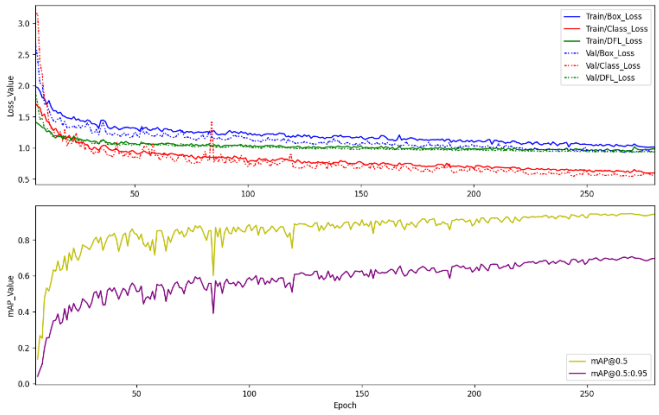

Fig. 3. Changes in loss and mAP values with epochs during training.

C. Evaluation Metrics

To evaluate the detection performance of the Lite-CropNet model, this study uses Precision (P), Recall (R), mean Average Precision (mAP@0.5, mAP@0.5:0.95), and F1 (F1-score) as evaluation metrics. A standard Intersection over Union (IoU) threshold of 0.5 was used in the experiment. If the overlap between a predicted bounding box and a labeled bounding box exceeds the IoU threshold, it is considered correct (true positive). Otherwise, the predicted bounding box is considered a false positive. When a labeled bounding box overlaps with a predicted bounding box below the threshold, it is considered a false negative. Precision (P) represents the proportion of correctly predicted objects by the model among all predicted objects. Recall (R) represents the proportion of correctly predicted objects by the model among all actual objects. Average Precision (AP) is defined as the area under the Precision-Recall (P-R) curve formed by the above Precision and Recall. The F1-score evaluates the model by balancing the weights of Precision and Recall. They are defined by the following equations, respectively [see Eq. (18) to Eq. (21)]:

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (18)$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (19)$$

$$mAP = \frac{1}{n}\sum_{1}^{n} P(R)d(R) \quad (20)$$

$$F1 = \frac{2\times Precision\times Recall}{Precision+Recall} \times 100\% \quad (21)$$

where, TP (True Positives), FP (False Positives), and FN (False Negatives) represent the number of true positives, false positives, and false negatives, respectively. In combination, "TP + FP" is the total number of detected targets, and "TP + FN" is the total number of real targets in the image. mAP@0.5 represents the average value of mAP when the IoU threshold is 0.5. Among them, mAP@0.5:0.95 represents the average value of mAP under different IoU thresholds (from 0.5 to 0.95, with a step size of 0.05). Since the dataset has three category labels for different growth stages, n=3.

V. Results and Analysis

A. Comprehensive Evaluation of Model Performance

To evaluate the superiority of the Lite-CropNet model, the Lite-CropNet model was compared with five advanced object detection methods in terms of detection performance, including CenterNet [41], Faster R-CNN [24], EfficientDet [42], FCOS [43], and DETR [23]. The same training set and test set were used to train and test these models, and hyperparameter tuning was performed for all methods to ensure fair and objective results. The comprehensive evaluation indicators of each model for the three growth stages in the object detection task are shown in Table II.

TABLE II. Comparison of Evaluation Metrics of Different Models

Model	P	R	F1	mAP@0.5	mAP@0.5:0.95
CenterNet	73.5%	66.4%	69.8%	48.0%	15.9%
Faster R-CNN	57.5%	58.4%	57.9%	59.0%	25.4%
EfficientDet	52.1%	51.0%	51.5%	47.4%	16.8%
FCOS	62.4%	81.6%	70.7%	73.4%	44.7%
DETR	63.5%	72.7%	67.8%	61.8%	27.2%
YOLOv8	72.3%	81.6%	76.0%	83.3%	54.2%
YOLOv11	**77.0%**	75.1%	76.6%	77.1%	51.7%
Lite-CropNet	75.4%	**86.2%**	**80.4%**	**85.7%**	**56.8%**

The experiments show that the comprehensive performance of Lite-CropNet is superior to other object detection methods. The FCOS model ranks second in comprehensive performance,

with relatively high accuracy and recall for object detection. However, due to its single and lightweight network structure, it may have certain difficulties in handling occluded tomatoes and tomatoes of different sizes. CenterNet adopts an object detection method based on center points. It may have poor overall performance because the model focuses too much on the target center point and ignores other possible target parts. The Faster R-CNN and EfficientDet models show balanced performance in the P and R indicators, but their overall performance is not excellent. It is worth noting that the DETR model is different from other object detection models. It adopts an end-to-end training method, which reduces cumbersome steps and manual intervention. However, it is easily limited by annotated data and computing resources, and has high computational costs, which is not suitable for real-time application scenarios, and does not achieve good performance in this task.

In summary, the Lite-CropNet model performs best in the tomato detection task. The advantages of Lite-CropNet may come from its strong adaptability to multi-scale, multi-growth stage, and occluded targets, as well as the appropriate feature decoding and optimization processing of the feature maps of the Encoder in the Decoder.

B. Lightweight Analysis of Different Models

When deploying the tomato detection and classification model to low-end edge devices, lightweight performance is a key factor. To comprehensively consider the lightweight performance of the model, two indicators—Frame Per Second (FPS) and model parameter quantity (Params)—were used to evaluate the lightweight effect of different models. The FPS indicator measures the number of frames that the model can process per unit time; a higher FPS value indicates that the model can process more input data in the same time and has higher real-time performance. Params reflects the number of model parameters; a smaller number of parameters means that the model is more lightweight, requires fewer computing resources during inference, and is directly related to the storage space of the model and the efficiency of computing resource usage. Their calculation methods are as follows [see Eq. (22) and Eq. (23)]:

$$FPS = \frac{1000}{pre-process+inference+NMS} \quad (22)$$

$$Params = i \cdot (k \cdot k) \cdot o + o \quad (23)$$

In the equations, is the input size, is the convolution kernel size, and is the output size. Among them, $pre-process$, $inference$, and Non-Maximum Suppression (NMS) represent the time required for preprocessing, inference, and non-maximum suppression of each image, respectively.

This experiment was conducted on a PC equipped with a relatively low-end Nvidia GTX 1650 GPU. Fig. 4 lists the comparison of lightweight parameters of different models. From the comparison results, the Lite-CropNet model achieves an excellent comprehensive level, with a high frame rate of 76.9 and lightweight parameters of only 4.4M. In contrast, although the EfficientDet model has a small number of parameters (15.0M), its frame rate is only 17.2, which is relatively low. The Faster R-CNN and FCOS models also have a large number of parameters. The DETR model has a frame rate of 21.3 and a parameter quantity of 158M; although it adopts an end-to-end training method, its performance is still relatively general here. The CenterNet model has a high frame rate, but its parameter quantity is still not sufficient for deploying lightweight models. Although YOLOv11 shows balanced performance in accuracy (mAP@0.5 is 77.1%) and frame rate (63.6FPS), its parameter quantity (6.3M) and computational complexity are still higher than those of Lite-CropNet. This further proves that the advantages of Lite-CropNet in comprehensive performance come from its lightweight design, providing a more feasible solution for real-time applications on low-end devices. Especially in the agricultural field where available resources are limited, lightweight models are usually more favored because they require fewer computing resources, which is important for some agricultural managers and can effectively reduce their economic burden.

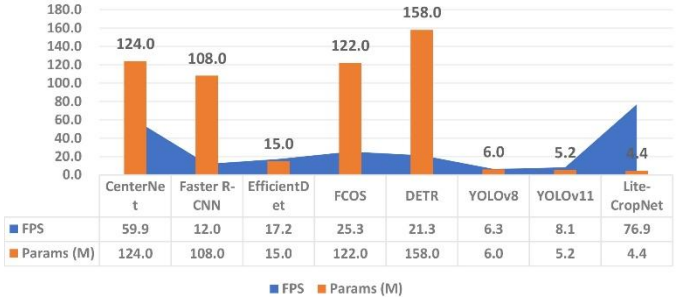

Fig. 4. FPS and Params comparison of different models.

C. Inference Speed Analysis of Different Models in a CPU Environment

In agricultural scenarios, many low-end edge devices (such as small embedded controllers and low-cost industrial computers) may not be equipped with GPUs and only rely on CPUs for inference. Therefore, a supplementary test on the inference speed of mainstream non-YOLO series models in the CPU environment was conducted. The experimental equipment was an Intel Celeron J4125 processor (4 cores and 4 threads, main frequency 2.0GHz) commonly used in agricultural scenarios, with 8GB of memory, and the test image resolution was still 640×640. The experimental data are shown in Table III.

TABLE III. COMPARISON OF EVALUATION METRICS OF DIFFERENT MODELS IN A GPU OR CPU ENVIRONMENT

Model	CPU FPS	GPU FPS (GTX 1650)	Parameter Count (M)
Lite-CropNet	**8.2**	**76.9**	**4.4**
CenterNet	6.5	59.9	108.0
Faster R-CNN	2.1	25.3	124.0
FCOS	3.3	21.3	122.0
DETR	1.8	17.2	158.0
EfficientDet	4.7	12.0	15.0

The results show that Lite-CropNet can still reach 8.2FPS in the CPU environment, which meets the basic requirements of real-time agricultural monitoring (usually requiring ≥5FPS). This performance benefits from its lightweight design:

compared with Faster R-CNN (2.1FPS) and DETR (1.8FPS), the parameter scale of Lite-CropNet is only 3.5% and 2.8% of the former, respectively, reducing memory occupation and computational delay. Compared with the same-level EfficientDet (4.7FPS), the CPU inference speed of Lite-CropNet is increased by 74%, which is due to the more concise feature fusion strategy adopted by its decoder, which reduces computational complexity. The Lite-CropNet in this study further optimizes the attention mechanism and loss function, and while maintaining accuracy, it achieves more efficient CPU inference, providing a feasible solution for agricultural scenarios without GPUs.

D. Visualization Analysis of Inference Results

By comparing the model prediction results with the real annotation results, the reasons affecting the detection performance of the model can be analyzed. Fig. 5 shows typical error cases, where the red arrows point to the regions missed by the model. Compared with other models, this model fails to correctly mark the real target regions. Missed detection is defined as failing to detect fruits with an occluded area of less than 50%, which is the standard for annotating the training set and test set in the TomatOD dataset.

From the experiment, it was observed that false detections mainly occur due to similar appearances, where leaves or debris are mistakenly identified as tomatoes. However, it is reassuring that in the test set, except for the case where the occlusion degree of the region pointed by the red arrow in the figure exceeds and is close to 50%, no missed detections were found in other images. Compared with the baseline model YOLOv11, Lite-CropNet achieves a lower missed detection rate in occluded scenarios (such as leaf occlusion and fruit overlap) and low-light environments. This indicates that the model can better capture tomato fruits under various environmental conditions, and even in dim environments with insufficient light, the model still shows similar discrimination ability to humans, further proving that the model has strong robustness and generalization.

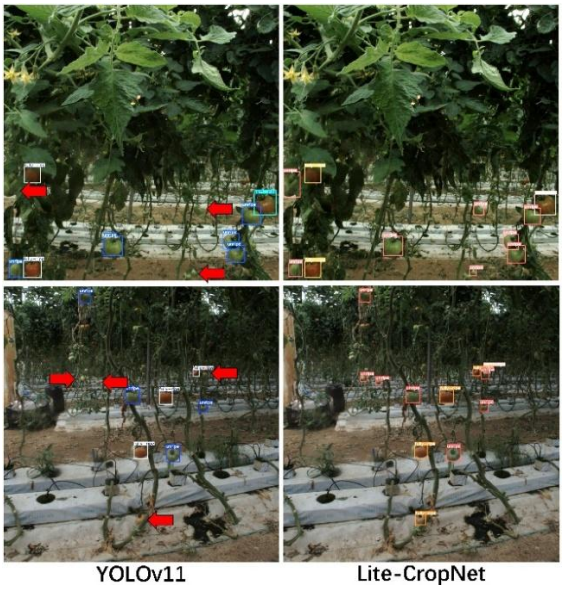

Fig. 5. Inference error results of the baseline model YOLOv11 and the Lite-CropNet model. Red arrows indicate missed detections.

E. Ablation Experiments

1) Ablation experiment on attention mechanism: In the above content, common attention mechanism methods are listed. To verify the detection differences of the Lite-CropNet model for tomatoes at each growth stage after adding different attention mechanisms to the upsampling and downsampling modules, ablation experiments were conducted using different attention mechanisms to analyze their performance differences.

The experimental results are shown in Table IV. The use of the Shuffle Attention mechanism has a positive impact on the model performance. In particular, the unique channel shuffling design of Shuffle Attention makes the mAP@0.5 indicator of the model at least 1.6% higher than that of other attention mechanisms. It can better capture global and local features, thereby comprehensively understanding the image content. Moreover, its mAP@0.5:0.95 ranks first at 57.4%, which is the only mechanism among all schemes that exceeds 87% in the core accuracy indicator. This design brings better-balanced performance in practical applications, enabling the model to achieve high levels in multiple key indicators.

TABLE IV. PERFORMANCE COMPARISON OF LITE-CROPNET USING DIFFERENT ATTENTION MECHANISMS

Attention	Precision (P)	Recall (R)	F1-Score (F1)	mAP@0.5	mAP@0.5:0.95
(None)	76.8%	80.7%	78.7%	85.5%	55.9%
SE[a]	78.4%	85.5%	**81.8%**	84.9%	56.9%
ECA[b]	80.2%	79.8%	80.0%	85.5%	56.2%
CBAM[c]	77.8%	79.3%	78.5%	83.8%	54.8%
Mlt_ECA[d]	75.4%	**86.2%**	80.4%	85.7%	56.8%
SA[e]	**80.4%**	79.5%	79.9%	**87.3%**	**57.4%**

[a] SE = Squeeze-and-Excitation

[b] ECA = Efficient Channel Attention

[c] CBAM = Convolutional Block Attention Module

[d] Mlt_ECA = Multi-scale Efficient Channel Attention

[e] SA = Shuffle Attention

In general, these observations indicate that the proposed Lite-CropNet model exhibits excellent robustness and generalization in practical tomato detection applications, providing a reliable automated solution for the agricultural field.

2) Ablation experiment on four-scale detection heads: To verify the adaptive value of the core innovation of Lite-CropNet—the "four-scale detection heads"—for crop targets of different sizes, this experiment constructs comparative models with only the number of detection heads modified. It focuses on analyzing the impact of the number of detection heads on the detection accuracy of small targets, large targets, and overall targets, so as to clarify the effectiveness of the scale design.

In this experiment, the core architecture of the model (CSPDarknet Encoder, Shuffle Attention, SIoU loss) was kept unchanged, and only the number of detection heads was adjusted: a two-scale model (2 detection heads, removing the

P2 small-scale and P5 large-scale branches) and a three-scale model (3 detection heads, removing the P5 large-scale branch) were constructed, and compared with the original four-scale model (4 detection heads, i.e., Lite-CropNet, retaining the full-scale branches of P2/P3/P4/P5). The performance comparison of the three groups of models on the same training set and validation set is shown in Table V.

TABLE V. PERFORMANCE COMPARISON OF MODELS WITH DIFFERENT NUMBERS OF DETECTION HEADS

Model Scale	P	R	F1	mAP@0.5	mAP@0.5:0.95
2	70.9%	82.2%	76.1%	78.2%	49.4%
3	72.7%	82.3%	77.2%	79.7%	50.7%
4 (Lite-CropNet)	**75.4%**	**86.2%**	**80.4%**	**85.7%**	**56.8%**

The experimental results show that the full-scale detection performance of the original four-scale model (4 detection heads) of Lite-CropNet is significantly better than that of the two-scale (2 detection heads) and three-scale (3 detection heads) comparative models: its overall detection accuracy (mAP@0.5 of 85.7%) and comprehensive performance index (F1 of 80.4%) are both at the optimal level. This proves that the four-scale detection heads can effectively improve the model's detection robustness for crop targets in the full scene by integrating feature information of different levels, including P2 (small-scale), P3 (medium-scale), P4 (medium-large scale), and P5 (large-scale), and reduce missed detections and false detections caused by insufficient scale adaptation. From the perspective of adaptive ability for segmented targets, the addition of the small-scale branch (P2) is the key to solving the problem of small fruit detection, which can effectively improve the model's recognition accuracy for small targets such as tomato young fruits and edge small fruits; while the large-scale branch (P5) can strengthen the positioning ability for large targets such as mature large tomatoes, avoiding bounding box deviation caused by the lack of global semantic information.

In summary, the four-scale detection heads enable Lite-CropNet to better adapt to multi-scale crops, solve the problem of insufficient detection ability of single-scale or few-scale models for targets of extreme sizes (such as extremely small young fruits and extra-large mature fruits), achieve a balance between accuracy and efficiency, and provide universal adaptive capabilities for multi-variety crop detection.

VI. DISCUSSION

In this research work, the Lite-CropNet model was proposed. Considering the cost of hardware resources, the lightweight design of the model was focused on. The TomatOD dataset was used to conduct in-depth research and performance analysis on the problem of crop fruit object detection and classification. Experiments have shown that while maintaining high accuracy, Lite-CropNet has a smaller model parameter size and higher frame rate, making it suitable for real-time application scenarios on low-end devices. In addition, the TomatOD dataset used was collected from real tomato planting scenarios, considering various challenges such as light changes and occlusions, resulting in a model with better robustness and generalization.

Furthermore, although the Lite-CropNet model generally performs well, the research still has potential shortcomings. First, a major problem is the relatively small scale of the dataset. It only contains 277 images and 2,413 tomato samples, and was collected from a single-region soilless cultivation greenhouse on Crete, Greece. It does not cover diverse planting environments (such as open fields and high-humidity solar greenhouses) in different countries and regions, nor does it include extreme scenarios such as rainstorm reflections and abnormal fruit appearances caused by diseases and pests, resulting in limited data diversity. Although the dataset already includes various environmental conditions and occlusion situations and successfully addresses them, the limited amount of data and the limitations of scene coverage may have a certain impact on the cross-region and cross-scene generalization ability and robustness of the model. Larger-scale datasets can usually better capture the diversity of the real world. As mentioned in the visual analysis above, the model inevitably has false detection problems when facing similar backgrounds, which is also considered to be caused by this reason. Second, for different tomato varieties in different countries and regions, the appearance characteristics of the plants are affected by various factors such as growth environment, soil conditions, and climate, and there may be slight differences in appearance, such as differences in fruit size and color depth, which may also affect the performance of the model. At the same time, although the study mentions that the model can be transferred to other horticultural crops such as potatoes and strawberries, it has not conducted verification for the morphological characteristics of these crops (such as clustered distribution of strawberries and underground fruiting of potatoes), and the multi-crop adaptability still needs further testing. This requires considering a wider and more diverse dataset during model training to ensure that the model has good adaptability to plant varieties in different countries and regions [44], [45].

VII. CONCLUSION

Efficient and accurate detection and counting of crop fruits have long been a challenging task. In this study, a novel Lite-CropNet model was proposed, with YOLOv11 as the baseline, aiming to address the key issues in horticultural crop detection and classification. A lightweight encoder was adopted and a concise yet efficient decoder was designed, which alleviates the problems of information loss and degradation. The exploration of different attention mechanisms revealed that the Shuffle Attention mechanism exerts a positive effect, helping to improve the model's performance. More importantly, efficiency enhancement was prioritized in the model design: a more lightweight architecture with a smaller parameter scale was developed, providing a cost-effective and highly efficient automated solution for agricultural production.

Experiments demonstrate that Lite-CropNet performs excellently under various lighting and occlusion conditions, exhibiting strong robustness and generalization. Notably, Lite-CropNet does not sacrifice efficiency at the expense of accuracy; instead, it achieves a balance between the two. Its

efficient performance in the CPU environment (8.2FPS) expands its application scenarios, making it particularly suitable for low-cost monitoring devices in greenhouses (e.g., Raspberry Pi-based embedded systems). In the future, CPU inference speed can be further improved through model quantization.

This work not only provides advanced technical support for the agricultural field but also offers insights for future research directions. The next phase of research will expand the dataset scale to cover more horticultural crop varieties, thereby enhancing the model's adaptability. Additionally, efforts will be made to further extend Lite-CropNet's performance advantages in the CPU environment and deploy it on low-cost monitoring devices in greenhouses, enabling it to better adapt to agricultural environments with extremely limited resources. Meanwhile, adversarial elements will be introduced to improve and optimize the object detection model, enhancing its robustness in complex scenarios. It is expected that this study will inspire more researchers to engage in agricultural intelligence and automation, and make more in-depth contributions to this field.

ACKNOWLEDGMENT

The research is supported by the Research Project of Software Engineering Institute of Guangzhou (No: KY202405).

REFERENCES

[1] F. Maureira, K. Rajagopalan, and C. O. Stöckle, "Evaluating tomato production in open-field and high-tech greenhouse systems," J. Cleaner Prod., vol. 337, pp. 130459, Apr. 2022.

[2] M. Afonso, H. Fonteijn, F. S. Fiorentin, D. Lensink, M. Mooij, N. Faber, R. Wehrens, "Tomato fruit detection and counting in greenhouses using deep learning," Front. Plant Sci., vol. 11, pp. 571299, Nov. 2020.

[3] L. Gong, M. Yu, S. Jiang, V. Cutsuridis, and S. Pearson, "Deep learning based prediction on greenhouse crop yield combined TCN and RNN," Sensors, vol. 21, no. 13, pp. 4537, Jul. 2021.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436-444, May 2015.

[5] A. M. Hasan, F. Sohel, D. Diepeveen, H. Laga, and M. G. Jones, "A survey of deep learning techniques for weed detection from images," Comput. Electron. Agric., vol. 184, pp. 106067, May 2021.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Adv. Neural Inf. Process. Syst., vol. 25, pp. 1097-1105, 2012.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Las Vegas, NV, USA, 2016, pp. 770-778.

[8] F. Waldner and F. I. Diakogiannis, "Deep learning on edge: Extracting field boundaries from satellite images with a convolutional neural network," Remote Sens. Environ., vol. 245, pp. 111741, Aug. 2020.

[9] J. Chen, J. Zhou, Q. Li, H. Li, Y. Xia, R. Jackson, J. Zhou, "CropQuant-Air: an AI-powered system to enable phenotypic analysis of yield- and performance-related traits using wheat canopy imagery collected by low-cost drones," Front. Plant Sci., vol. 14, pp. 1219983, Apr. 2023.

[10] V. Bathini and K. U. Rani, "A review of analyzing different agricultural crop yields using artificial intelligence," Int. J. Adv. Comput. Sci. Appl., vol. 16, no. 1, pp. 1-10, 2025.

[11] V. Tsironis, S. Bourou, and C. Stentoumis, "tomatOD: Evaluation of object detection algorithms on a new real-world tomato dataset," in ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., vol. XLIII-B3-2020, pp. 1077-1084, 2020.

[12] S. Ahmad, Z. Chen, S. Ikram, and A. Ikram, "AI-enabled vision transformer for automated weed detection: Advancing innovation in agriculture," Int. J. Adv. Comput. Sci. Appl., vol. 15, no. 12, pp. 1-9, 2024.

[13] Y. Zhang, Y. Xu, J. Hou, and Y. Song, "LMS-YOLO11n: A lightweight multi-scale weed detection model," Int. J. Adv. Comput. Sci. Appl., vol. 16, no. 1, pp. 1-8, 2025.

[14] R. Reedha, E. Dericquebourg, R. Canals, and A. Hafiane, "Transformer neural network for weed and crop classification of high resolution UAV images," Remote Sens., vol. 14, no. 3, pp. 592, Feb. 2022.

[15] J. M. López-Correa, H. Moreno, A. Ribeiro, and D. Andújar, "Intelligent weed management based on object detection neural networks in tomato crops," Agronomy, vol. 12, no. 12, pp. 2953, Dec. 2022.

[16] Y. Mu, T. S. Chen, S. Ninomiya, and W. Guo, "Intact detection of highly occluded immature tomatoes on plants using deep learning techniques," Sensors, vol. 20, no. 10, pp. 2984, May 2020.

[17] F. Su, Y. Zhao, G. Wang, P. Liu, Y. Yan, and L. Zu, "Tomato maturity classification based on SE-YOLOv3-MobileNetV1 network under nature greenhouse environment," Agronomy, vol. 12, no. 7, pp. 1638, Jul. 2022.

[18] U. F. Rahim and H. Mineno, "Tomato flower detection and counting in greenhouses using faster region-based convolutional neural network," J. Image Graph., vol. 8, no. 4, pp. 107-113, Dec. 2020.

[19] U. F. Rahim, T. Utsumi, and H. Mineno, "Deep learning-based accurate grapevine inflorescence and flower quantification in unstructured vineyard images acquired using a mobile sensing platform," Comput. Electron. Agric., vol. 198, pp. 107088, Jul. 2022.

[20] H. K. Suh, J. Ijsselmuiden, J. W. Hofstee, and E. J. van Henten, "Transfer learning for the classification of sugar beet and volunteer potato under field conditions," Biosyst. Eng., vol. 174, pp. 50-65, Oct. 2018.

[21] D. K. Nkemelu, D. Omeiza, and N. Lubalo, "Deep convolutional neural network for plant seedlings classification," arXiv:1811.08404, Nov. 2018, unpublished.

[22] V. Tsironis, S. Bourou, and C. Stentoumis, "Tomatod: evaluation of object detection algorithms on a new real-world tomato dataset," Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., vol. 43, pp. 1077-1084, 2020.

[23] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in Proc. Eur. Conf. Comput. Vis., Glasgow, U.K., 2020, pp. 213-229.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," Adv. Neural Inf. Process. Syst., vol. 28, pp. 91-99, 2015.

[25] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv:2004.10934, Apr. 2020, unpublished.

[26] C. Y. Wang, H. Y. M. Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops, Seattle, WA, USA, 2020, pp. 390-391.

[27] G. Jocher et al., "ultralytics/yolov5: v6.0 — YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support," Zenodo, Oct. 2021, doi: 10.5281/zenodo.5563715.

[28] Y. Sun, G. Chen, T. Zhou, Y. Zhang, and N. Liu, "Context-aware cross-level fusion network for camouflaged object detection," arXiv:2105.12555, May 2021, unpublished.

[29] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," Neural Netw., vol. 107, pp. 3-11, Nov. 2018.

[30] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in Proc. 14th Int. Conf. Artif. Intell. Stat., Fort Lauderdale, FL, USA, 2011, pp. 315-323.

[31] J. Xu, Z. Li, B. Du, M. Zhang, and J. Liu, "Reluplex made more practical: Leaky ReLU," in Proc. IEEE Symp. Comput. Commun., Rennes, France, 2020, pp. 1-7.

[32] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA, 2018, pp. 7132-7141.

[33] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in Proc. Eur. Conf. Comput. Vis., Munich, Germany, 2018, pp. 3-19.

[34] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Seattle, WA, USA, 2020, pp. 11534-11542.

[35] Q. L. Zhang and Y. B. Yang, "SA-Net: Shuffle attention for deep convolutional neural networks," in Proc. IEEE ICASSP, Toronto, ON, Canada, 2021, pp. 2235-2239.

[36] Z. Yu, J. Ye, C. Li, H. Zhou, and X. Li, "TasselLFANet: A novel lightweight multi-branch feature aggregation neural network for high-throughput image-based maize tassels detection and counting," Front. Plant Sci., vol. 14, pp. 1158940, May 2023.

[37] Z. Gevorgyan, "SIoU loss: More powerful learning for bounding box regression," arXiv:2205.12740, May 2022, unpublished.

[38] D. Tzutalin, "LabelImg: Graphical image annotation tool," GitHub. https://github.com/tzutalin/labelImg, 2022.

[39] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," Adv. Neural Inf. Process. Syst., vol. 32, pp. 8024-8035, 2019.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, Dec. 2014, unpublished.

[41] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," in Proc. IEEE/CVF Int. Conf. Comput. Vis., Seoul, Korea, 2019, pp. 6569-6578.

[42] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Seattle, WA, USA, 2020, pp. 10781-10790.

[43] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in Proc. IEEE/CVF Int. Conf. Comput. Vis., Seoul, Korea, 2019, pp. 9627-9636.

[44] J. Ye, Z. Yu, Y. Wang, D. Lu, and H. Zhou, "WheatLFANet: In-field detection and counting of wheat heads with high-real-time global regression network," Plant Methods, vol. 19, no. 1, pp. 103, Dec. 2023.

[45] M. Gatto et al., "Trends in varietal diversity of main staple crops in Asia and Africa and implications for sustainable food systems," Front. Sustain. Food Syst., vol. 5, pp. 626714, Mar. 2021.